IN THE CLAIMS:

Please amend claims 1, 10, and 24, and add new claims 25-51.

1. **(CURRENTLY AMENDED)** A computer-implemented system for communicating among a plurality of computer-based machines interconnected through a network, comprising:

a first group of computer modules operating on a first machine;

a second group of computer modules operating on a second machine; and

a data network ring connected to the first group and the second group, said modules of the first and second groups exchanging messages over the data network ring,

wherein any one of the modules in the first group may operate as a proxy that is responsible for messages sent to or from the first machine over the data network ring,

wherein any one of the modules in the second group may operate as a proxy that is responsible for messages sent to or from the second machine over the data network ring;

wherein the module operating as a proxy on the first machine is responsible for messages involving another module operating on the first machine;

wherein the module operating as a proxy on the second machine is responsible for messages involving another module operating on the second machine.

2. **(ORIGINAL)** The system of claim 1 wherein the modules of the first and second groups handle data views.

3. **(ORIGINAL)** The system of claim 2 further comprising:

a message that is created by a module in the first group and sent to the proxy of the first group,

said first group's proxy sending the message over the data network ring to the proxy of the second group,

said second group's proxy sending the message to the other modules in the second group.

4. **(ORIGINAL)** The system of claim 1 further comprising:

a distributed message ring class that provides methods and data structures by which the proxies of the first and second groups are to operate as proxies, said proxies of the first and second groups being instantiations of the distributed message ring class.

5. **(ORIGINAL)** The system of claim 4 wherein the first machine has a first data network ring instance, and the second machine has a second data network ring instance, wherein the first and second ring instances form the data network ring.

6. **(ORIGINAL)** The system of claim 5 wherein each module in the first and second groups has a logical interface to the data network ring.

7. **(ORIGINAL)** The system of claim 6 wherein each module in the first and second groups are equal members of the data network ring.

8. **(ORIGINAL)** The system of claim 7 wherein each module in the first and second groups share any overhead to implement the data network ring.

9. **(ORIGINAL)** The system of claim 1 further comprising:

a cooperative proxy election mechanism wherein another module of the first group is selected as a new proxy for the first group based upon the proxy of the first group being closed.

10. **(CURRENTLY AMENDED)** The system of claim 9 wherein proxy election for the first group is solely contained and conducted by the modules in the first group;

wherein the first group includes only a single proxy that is responsible for messages involving the first group.

11. **(ORIGINAL)** The system of claim 1 further comprising:

a cooperative proxy election mechanism wherein another module of the first group is selected as a new proxy for the first group based upon the proxy of the first group being deconstructed.

12. **(ORIGINAL)** The system of claim 1 further comprising:

a threading mechanism used by the proxies of the first and second groups to handle messages sent to their respective machines.

13. **(ORIGINAL)** The system of claim 12 wherein messages are sent over the data network ring asynchronously so that the sending module does not block the hosting thread of execution.

14. **(ORIGINAL)** The system of claim 1 wherein the computer modules are objects with methods and data structures.

15. **(ORIGINAL)** The system of claim 1 wherein ultimate source of a message sent over the data network ring is unknown by ultimate recipient of the message.

16. **(ORIGINAL)** The system of claim 1 wherein the modules in the first and second groups communicate among themselves anonymously.

17. **(ORIGINAL)** The system of claim 1 wherein the modules in the first and second groups communicate among themselves at a peer-to-peer level.

18. **(ORIGINAL)** The system of claim 1 wherein the modules in the first and second groups communicate among themselves at an object level.

19. **(ORIGINAL)** The system of claim 1 wherein scope of a data network ring is based upon a pre-selected level.

20. **(ORIGINAL)** The system of claim 19 wherein the preselected level is selected from the group consisting of process level, machine level, and network level.

21. **(ORIGINAL)** The system of claim 1 wherein at least a portion of the first group's modules are used within a web browser.

22. **(ORIGINAL)** The system of claim 1 wherein at least a portion of the first group's modules are multiple instances of an application running.

23. **(ORIGINAL)** The system of claim 1 wherein process-level status propagation messages are sent over the data network ring.

24. **(CURRENTLY AMENDED)** A computer-implemented method for communicating among a plurality of computer-based machines interconnected through a network, comprising the steps of:

operating a first group of computer modules on a first machine;

operating a second group of computer modules on a second machine;

exchanging messages between the first group and second group of modules over a data network ring, wherein any one of the modules in the first group may operate as a proxy that is responsible for messages sent to or from the first machine over the data network ring, wherein any one of the modules in the second group may operate as a proxy that is responsible for messages sent to or from the second machine over the data network ring; ~~and~~

wherein the module operating as a proxy on the first machine is responsible for messages involving another module operating on the first machine;

wherein the module operating as a proxy on the second machine is responsible for messages involving another module operating on the second machine~~——~~

~~electing a module in the first group to serve as a new proxy for the first group based upon the proxy of the first group being closed, wherein proxy election for the first group is solely contained and conducted by the modules in the first group~~.

25. **(NEW)** The method of claim 24 wherein the modules of the first and second groups handle data views.

26. **(NEW)** The method of claim 25 further comprising:

a message that is created by a module in the first group and sent to the proxy of the first group,

said first group's proxy sending the message over the data network ring to the proxy of the second group,

said second group's proxy sending the message to the other modules in the second group.

27. **(NEW)** The method of claim 24 further comprising:

using a distributed message ring class that provides methods and data structures by which the proxies of the first and second groups are to operate as proxies,

said proxies of the first and second groups being instantiations of the distributed message ring class.

28. **(NEW)** The method of claim 27 wherein the first machine has a first data network ring instance, and the second machine has a second data network ring instance, wherein the first and second ring instances form the data network ring.

29. **(NEW)** The method of claim 28 wherein each module in the first and second groups has a logical interface to the data network ring.

30. **(NEW)** The method of claim 29 wherein each module in the first and second groups are equal members of the data network ring.

31. **(NEW)** The method of claim 30 wherein each module in the first and second groups share overhead to implement the data network ring.

32. **(NEW)** The method of claim 24 wherein another module of the first group is selected as a new proxy for the first group based upon the proxy of the first group being closed.

33. **(NEW)** The method of claim 32 wherein proxy election for the first group is solely contained and conducted by the modules in the first group;

wherein the first group includes only a single proxy that is responsible for messages involving the first group.

34. **(NEW)** The method of claim 24 wherein another module of the first group is selected as a new proxy for the first group based upon the proxy of the first group being deconstructed.

35. **(NEW)** The method of claim 24 wherein a threading mechanism is used by the proxies of the first and second groups to handle messages sent to their respective machines.

36. **(NEW)** The method of claim 35 wherein messages are sent over the data network ring asynchronously so that the sending module does not block the hosting thread of execution.

37. **(NEW)** The method of claim 24 wherein the computer modules are objects with methods and data structures.

38. **(NEW)** The method of claim 24 wherein ultimate source of a message sent over the data network ring is unknown by ultimate recipient of the message.

39. **(NEW)** The method of claim 24 wherein the modules in the first and second groups communicate among themselves anonymously.

40. **(NEW)** The method of claim 24 wherein the modules in the first and second groups communicate among themselves at a peer-to-peer level.

41. **(NEW)** The method of claim 24 wherein the modules in the first and second groups communicate among themselves at an object level.

42. **(NEW)** The method of claim 24 wherein scope of a data network ring is based upon a pre-selected level.

43. **(NEW)** The method of claim 42 wherein the preselected level is selected from the group consisting of process level, machine level, and network level.

44. **(NEW)** The method of claim 24 wherein at least a portion of the first group's modules are used within a web browser.

45. **(NEW)** The method of claim 24 wherein at least a portion of the first group's modules are multiple instances of an application running.

46. **(NEW)** The method of claim 24 wherein process-level status propagation messages are sent over the data network ring.

47. **(NEW)** The method of claim 24 wherein the modules of the first group perform non-proxy operations;

wherein the proxy module of the first group operates in role of a proxy as well as performs non-proxy operations.

48. **(NEW)** The method of claim 47 wherein a non-proxy operation includes a data view operation.

49. **(NEW)** A computer-implemented system for communicating among a plurality of computer-based machines interconnected through a network, comprising the steps of:

means for operating a first group of computer modules on a first machine;

wherein a second group of computer modules operates on a second machine;

means for exchanging messages between the first group and second group of modules over a data network ring, wherein any one of the modules in the first group may operate as a proxy that is responsible for messages sent to or from the first machine over the data network ring, wherein any one of the modules in the second group may operate as a proxy that is responsible for messages sent to or from the second machine over the data network ring; and

means for operating a module of the first group as a proxy on the first machine such that the proxy is responsible for messages involving another module operating on the first machine.

50. **(NEW)** The system of claim 1 wherein the modules of the first group perform non-proxy operations;

wherein the proxy module of the first group operates in role of a proxy as well as performs non-proxy operations.

51. **(NEW)** The system of claim 50 wherein a non-proxy operation includes a data view operation.